Memo/Reply From
JOSHUA LEDERBERG

TO: Thos. R. Shroffer
W. S. J.

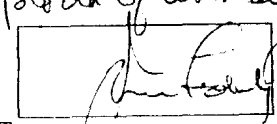
NOV - 2 1982

How smart is DENDRAL

As one of the authors of DENDRAL I was naturally pleased about the attributions — Thank you!

However, the remark that "Dendral is better than any analytical chemist...." goes a mite further than any claim I can remember making for it; and I expect you are in a chain of indirect references that has sheared off the footnotes and qualifiers. I am still proud of what Dendral can do.

Yours,



Joshua Lederberg
President
The Rockefeller University
New York, N.Y. 10021
(212) [redacted]

☐ ORIG. RETAINED
☐ COPY

— REPLY FORM —

McGraw-Hill Book Company

1980

New York St. Louis San Francisco Auckland Bogotá Hamburg Johannesburg
Lisbon London Lucerne Madrid Mexico Montreal New Delhi Panama
Paris San Juan São Paulo Singapore Sydney Tokyo Toronto

President, The Rockefeller University
Formerly, Professor of Genetics
Stanford University

Joshua Lederberg

Professor of Computer Science
Stanford University

Edward A. Feigenbaum

Adjunct Professor of Computer Science
Stanford University

Bruce G. Buchanan

Research Scientist
University of Michigan

Robert K. Lindsay

The DENDRAL Project

APPLICATIONS OF ARTIFICIAL INTELLIGENCE FOR ORGANIC CHEMISTRY

   d. Bond types (single, double, triple, aromatic)
   e. How to detect topological symmetry
   f. How to compute degree of unsaturation from empirical formula
   g. How to draw reasonable planar projections of molecular structures
   h. How to generate all isomers *including fused rings, spiro forms*, etc.
   i. How to generate all stereoisomers
   j. How to find cycles and arbitrarily complex subgraphs
   k. How to find the greatest common subgraph among a set of graphs
   l. How to label nodes and edges of graphs in all distinct ways, taking account of symmetry
   m. How to simulate specific chemical transformations, such as synthetic reactions
2. Knowledge of chemical stability
   a. Twenty classes of unstable acyclic structures known; any others can be specified
   b. How to recognize keto-enol tautomerism; other tautomers can be specified
   c. Terpene rule
   d. Isoprene rule
   e. Bredt's rule
3. Knowledge of mass spectrometry
   a. How to infer the formula of any molecular ion
   b. How to compute results of any specified fragmentation and rearrangement
   c. How to predict metastable peaks and use them for confirmation of inferences
   d. Rule of charge placement on fragments (but not on atoms)
   e. Half-order theory produces rough prediction of actual spectra
   f. Refined theory can be added for any family of structures (now available for ketones, ethers, alcohols, amines, thiols, thioethers, estrogenic steroids, keto-androstanes, marine sterols, and aromatic acids)
   g. McLafferty rearrangement, water elimination, carbon monoxide elimination, carbon dioxide elimination, and elimination of other user-defined "neutral species"
   h. Distinguishes high- and low-resolution spectra
   i. Distinguishes *low voltage and high voltage measurements*
4. Knowledge *not* available to DENDRAL
   a. Three-dimensional structure (except of stereoisomerism)
   b. Polymeric structures
   c. Quantum mechanical explanations of mass spectrometry processes
   d. Electronegativity
   e. Physical properties such as dipole moment, molecular susceptibility, melting point, crystal structure, and many others

In addition, all the knowledge of LISP is presupposed by the DENDRAL programs. For example, arithmetic and set theoretic operations, symbol manipulation, interpretation of complex procedures, and countless bookkeeping operations. Considerable amounts of code are devoted to keeping track of intermediate results in the overall processing. This "*specialized bookkeeping*" knowledge is not very profound, yet it is indispensable for the integration of many complex procedures.

Almost all DENDRAL's knowledge is tailored to the task of molecular structure elucidation. In spite of the elegance and simplicity of computing concepts we have to work with, the problem-solving procedures in DENDRAL are still very special purpose, complex, and voluminous. Making the procedures, and knowledge base, more general would have increased the burden of debugging them in most cases.[1]

9.2.2 How Is Knowledge Employed in Heuristic DENDRAL and Meta-DENDRAL?

In Chapter 3 we described the organization of problem-solving systems. The major bifurcation was between algorithms and heuristic programs. Heuristic programs were further characterized as either search through a *space of subproblems* or as generation of candidates from a *space of potential solutions*. It is also possible that a problem space consists of *partial solutions, as in the* HEARSAY *speech-understanding system*, [Erman and Lesser (1975), among others]. A space of partial solutions is in effect a combination of a space of solutions and a space of subproblems.

In either case, search or generation, the alternatives considered may be limited to those known to be legal according to a given rule, or to those that are merely plausible. In each case, problem solving may terminate with the discovery of a satisfactory solution, or proceed until the optimal solution is found.

To this initial division of heuristic programs, we added the concept of planning as exemplified by the planning phase of DENDRAL's basic solution generation method. *More generally, the use of planning can lead to computational economies in two ways.* Planning can *prune* the space (of subproblems or solutions) by eliminating sections of it and directing search to certain other sections, or it can *guide* the problem solving, either by ordering the search/generation sequence, or by modifying one subproblem/solution candidate to produce the next, using a hill-climbing (evolutionary) method. In the hybrid case of a space of partial solutions, both pruning and guidance methods are applicable.

Planning, furthermore, may be characterized along a different dimension. It may be data driven or expectation-driven. Data-driven planning begins by examination of data (from instruments, perhaps) and attempts to induce hypotheses to account for it. Expectation-driven planning begins with a model of the phenomenon and uses it to establish expectations.

It is possible to establish a taxonomy of problem-solving systems based on these *four binary characteristics*: (1) *subproblem* space versus solution space, (2) legal versus plausible alternatives considered, (3) pruning heuristics versus guidance heuristics, and (4) data driven versus expectation-driven. While such a taxonomy is useful, it should be remembered that the values of these dimensions are not mutually exclusive nor exhaustive, and indeed greater power will probably derive from future systems that combine these methods in various ways.

Heuristic DENDRAL may be characterized as a *generator* of *legal* solutions with

[1] For these reasons, we look forward to advances in automatic programming that will simplify the programming, debugging, and interactions of complex procedures.